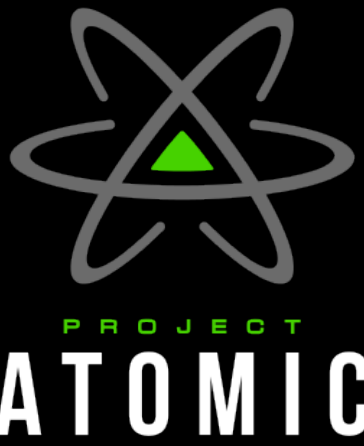


A Gentle Introduction to Linux Containers

Brian Proffitt
@TheTechScribe
@projectatomic
http://projectatomic.io

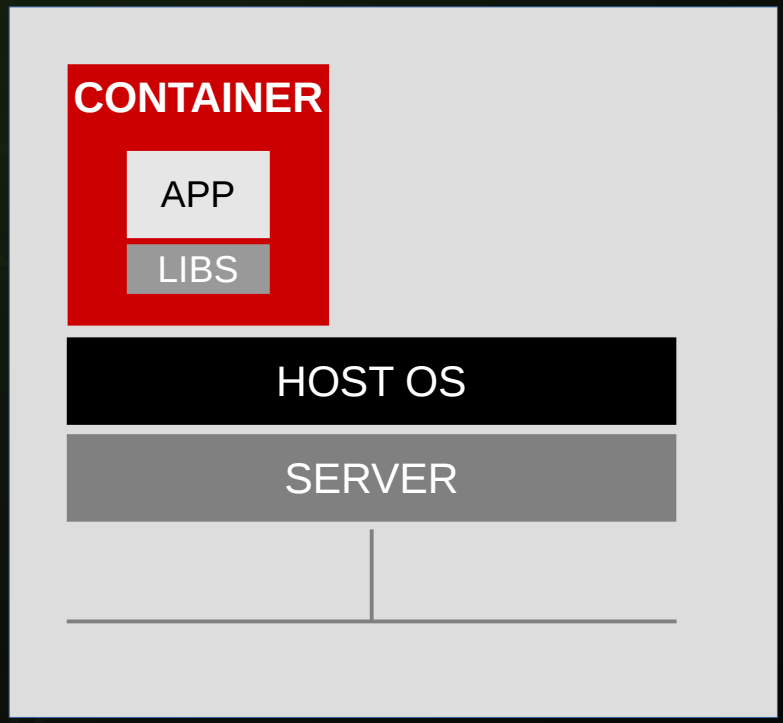
Introduction

- What are Linux Containers?
- Containers vs. Virtualization
- Benefits of Containers
- A Quick Container Run-Through
- Orchestration and Containers
- Creating Dockerfiles
- Building Images
- Learn More!

PROJECT ATOMIC

What Are Linux Containers?

- User space only – the image runs on the host kernel
- Includes application + runtime dependencies
- Must be the same as the host OS
- Isolates applications on the host OS
  - Uses Control Groups (cgroups), kernel namespaces, SELinux



PROJECT ATOMIC

Containers vs. Virtualization

- Containers are lightweight
  - Containers should be smaller and require fewer resources
- Require the underlying OS to be the same
  - (***e.g.*** can't run Linux containers natively on Windows)
- Containers are meant for “microservices”
  - A container may “live” just for a second or two

PROJECT ATOMIC

Container History

- Containers aren't new (nor only Docker):
  - Isolated user space instances go back to 1982 (at least) if you start from chroot
  - FreeBSD Jails: 1998
  - Linux-VServer: 2001
  - Virtuozzo: 2001 (OpenVZ - 2005)
  - LXC: 2008
  - **Docker: 2013 (based on LXC until verion 0.9 and libcontainer)**
  - systemd-nspawn: 2013
  - rkt: 2014

PROJECT ATOMIC

Why Linux Containers?

- Portability across environments
- Consistency from dev to test to production
- Faster/less overhead than VMs
- Layered – You can build on other people's work
- Modular – an app can be composed of multiple containers
- Dealing with legacy will be much, much easier

PROJECT ATOMIC

Docker Basics

Pulling an Image

- You can automatically pull an image from Docker Hub
- Be specific: unless you want **all** CentOS images, use the tags to specify which version
- docker pull centos:centos7
- docker pull fedora:24

PROJECT ATOMIC

Running an Image

- Once an image is on your system, it's dead easy to run
- You can run multiple versions of an image simultaneously
- Stopping an image preserves its state as a container
- docker run -ti fedora:24 /bin/bash
- docker start pensive_stallman

PROJECT ATOMIC

Containers vs. Images

- Image defines the filesystem for running an isolated Linux process (application)
- Container is a running instance of a Docker image with its own filesystem, network, and process spaces
- Remains a container even when stopped until it's deleted or committed to a new image

PROJECT ATOMIC

Image History

- Docker allows you to see the history of an image w/ docker history
- Only shows commits to the image, doesn't provide *container* history
- docker inspect image -- will show low-level information about a container in JSON format

PROJECT ATOMIC

Building an Image

- Two ways to build an image:
  - Interactively
  - Dockerfiles
- Assumptions:
  - You're using a base image (e.g. Fedora, CentOS, Debian) rather than from scratch
  - You want to automate as much as possible

PROJECT ATOMIC

Interactively

- Start with the base image
- docker run -ti centos:centos7 /bin/bash
- [do stuff]
- docker commit -a="you" -m="message" \
  a42dffa1 user/image:tag
- docker history user/image:tag
- Rinse, repeat as necessary

PROJECT ATOMIC

Saving/Exporting

- To share the image you can save or export the image
- Save: will save an image to a tarball with parent layers, tags, etc.
- Export: “flattens” the image to go to a tarball
- Export: lose history, but also provides a smaller image

PROJECT ATOMIC

Loading/Importing

- You can grab another Docker image tarball and use it on your system
- docker load will import a tarball and restore images and tags
- docker import creates a new, untagged image w/out the history

PROJECT ATOMIC

Push to Registry

- Instead of pushing tarballs around, you can use the Docker Hub or a local registry
- Like “git” you use “pull” to bring down content, and “push” to move content to the repo
- Can easily run local registries if your content doesn't belong on Docker Hub

PROJECT ATOMIC

Starting with Dockerfiles

- Grab any Dockerfile:
  - CentOS:
    `https://github.com/CentOS/CentOS-Dockerfiles/`
  - Fedora:
    `https://github.com/fedora-cloud/Fedora-Dockerfiles`
- Run it:

```
docker build --rm -t $USER/app:tagname .
```

PROJECT ATOMIC

Creating Dockerfiles

- Dockerfiles can be ***very*** simple or fairly complex (depending on your needs)
- One service per container (e.g. httpd in one, PostgreSQL in another)
- Thousands upon thousands of Dockerfiles to read and experiment with

PROJECT ATOMIC

Creating Dockerfiles

- Basic format:

  FROM Acme Project

  MAINTAINER Wile E. Coyote

  RUN yum update && yum clean all

  ADD localfile /etc/localfile.conf

  EXPOSE 80

  CMD [“/run-httpd.sh”]

PROJECT ATOMIC

Demo

PROJECT ATOMIC

Thanks! Questions?

Speaker
@twitter_user

email@redhat.com
@projectatomic

PROJECT ATOMIC